



# Styles en LitElement



Miguel Angel Alvarez  
@midesweb



**Miguel Angel Alvarez**

**miguel@desarrolloweb.com**

**@midesweb**

# En esta sesión:

- Maneras de definir hojas de estilo
- Selector :host
- Estilos en slots
- Aplicar estilo desde fuera del componente
- Personalización (theming) con variables CSS



Miguel Angel Alvarez  
@midesweb



# STYLES



Miguel Angel Alvarez  
@midesweb

# Definición de estilos

Existen diversas maneras de incorporar estilos:

1. La etiqueta <style> en el template
2. Propiedad estática styles
3. En una hoja de estilos externa
4. En el atributo style de una etiqueta



Miguel Angel Alvarez  
@midesweb

# 1.- Etiqueta <style>

Estilos aplicados en el template definido en el método render(), por medio de la etiqueta <style>.

```
render() {  
    return html`  
        <style>  
            p {  
                color: red;  
            }  
        </style>  
        <p>Bienvenidos a ${this.donde}!!!</p>  
    `;  
}
```



Miguel Angel Alvarez  
@midesweb

## 2.- Propiedad static styles

Estilos aplicados mediante una propiedad estática.

Permite tener una definición de estilos única, que reutilizan todos los componentes de un tipo. **Aumenta el rendimiento!**

\* Basada en Constructable Stylesheets:

<https://developers.google.com/web/updates/2019/02/constructable-stylesheets>

```
static get styles() {  
  return css`  
    div {  
      display: flex;  
      align-items: center;  
      margin-bottom: 8px;  
    }  
    .completed span {  
      text-decoration: line-through;  
      color: #888;  
    }  
    span {  
      margin-left: 5px;  
    }  
  `;  
}
```



Miguel Angel Alvarez  
@midesweb

## 2.- Propiedad static styles

Estilos aplicados mediante una propiedad estática.

Es posible devolver un array de declaraciones CSS, de este modo podemos sumar estilos al componente que nos vienen de otros lugares.

\* Basada en Constructable Stylesheets:

<https://developers.google.com/web/updates/2019/02/constructable-stylesheets>

```
static get styles() {  
  return css`  
    div {  
      display: flex;  
      align-items: center;  
      margin-bottom: 8px;  
    }  
    .completed span {  
      text-decoration: line-through;  
      color: #888;  
    }  
    span {  
      margin-left: 5px;  
    }  
  `;  
}
```



Miguel Angel Alvarez  
@midesweb



## 3.- Hojas de estilo externas

Usando la etiqueta <link>, igual que se cargan en el contexto de un sitio web normal.

- Desaconsejado
- Puedes necesitarlo si trabajas con preprocesadores (ej. Sass)
- Los estilos pueden tardar en renderizarse. Verías el componente sin estilos unos instantes durante la carga.
- Trabajar con URLs no es viable cuando trabajas en un componente que piensas publicar



Miguel Angel Alvarez  
@midesweb

# 4.- Atributo style de una etiqueta

Puedes usar el atributo style de cualquier tag HTML

- Usar como último recurso (ni lo sugieren en la documentación)
- Puede ser necesario para algunos componentes en algunas situaciones y navegadores específicos. (Edge)

```
render() {  
  return html`  
    <div  
      id="img"  
      role="img"  
      style="background-image: url('${this._assignImage(this.src)}'); border-radius: ${this.si  
    ></div>  
  `;  
}
```



Miguel Angel Alvarez  
@midesweb



# SELECTOR

## :host



Miguel Angel Alvarez  
@midesweb

# :host

Con este selector aplicamos estilo a la etiqueta "host", la que envuelve al componente.

## :host

Afecta a todo host.

## :host(selector)

Afecta a el host que aplique tal selector

```
static get styles() {  
  return css`  
    :host {  
      display: block;  
      border-radius: 5px;  
      padding: 10px;  
      box-shadow: 2px 2px 9px #ccc;  
      margin-bottom: 15px;  
    }  
    :host(.background-white) {  
      background-color: #fff;  
    }  
    h1 {  
      font-weight: 300;  
    }  
  `;  
}
```



Miguel Angel Alvarez  
@midesweb

# Buenas prácticas con :host

- Colocar un display adecuado al caso (default inline)
- Aplicar estilo cuando el componente tiene el atributo hidden.

```
:host {  
  display: block;  
}  
  
:host([hidden]) {  
  display: none;  
}
```



Miguel Angel Alvarez  
@midesweb



# ESTILOS EN slots



Miguel Angel Alvarez  
@midesweb

# ¿Donde aplicar estilos a Slots?

Los puedes aplicar fuera de los componentes. Para el CSS es como si el slot no perteneciera al propio componente. Está fuera del shadow DOM, por lo que los estilos CSS del componente no le afectan.

```
<card-box title="Mi tarjeta">
  <p>
    Este es el contenido de mi tarjeta!
  </p>
  <nav>
    <a href="#">Uno</a> | <a href="#">Dos</a> | <a href="#">Tres</a>
  </nav>
</card-box>
```



Miguel Angel Alvarez  
@midesweb

# Binding a evento

Podemos llegar a aplicar unos estilos en los slots con el selector `::slotted()`.

Aplicar estilo al slot, a todas las etiquetas hijas directas del slot.

Aplicar estilo a los párrafos que hay en un slot, a todas las etiquetas P hijas directas del slot.

```
::slotted(*) {  
  border: 1px solid blue;  
}  
  
::slotted(p) {  
  background-color: red;  
}
```



Miguel Angel Alvarez  
@midesweb





# ESTILOS DESDE fuera del componente



Miguel Angel Alvarez  
@midesweb

# Estilos al host

Podemos aplicar estilos CSS a la etiqueta host de un componente.

Pero no podemos aplicar (desde fuera) estilos CSS a partes internas del componente.

\* Shadow DOM

```
<head>
  <title>Clase 3 estilos</title>
  <style>
    body {
      font-family: sans-serif;
      background-color: #eee;
    }
    card-box {
      margin: 15px 50px;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <h1>Clase 3</h1>
```



Miguel Angel Alvarez  
@midesweb

# Estilos al host

Podemos aplicar estilos a la raíz del componente.

Eso es una verdad a medias, porque hay ciertos estilos que sí penetran dentro del componente. Por ejemplo font-size, font-family, color.

\* comportamiento normal de CSS

Pero no podemos aplicar (desde fuera) estilos CSS a partes internas del componente.

\* Shadow DOM

```
<head>
  <title>Clase 3 estilos</title>
  <style>
```

```
    margin: 15px 50px;
    cursor: pointer;
  }
</style>
```

```
</head>
<body>
  <h1>Clase 3</h1>
```



Miguel Angel Alvarez  
@midesweb



# THEMING de componentes



Miguel Angel Alvarez  
@midesweb

# Theming de componentes

Utilizando variables CSS (**custom properties**) podemos conseguir definir el estilo de partes del componente.

Como desarrolladores de componentes debemos habilitar el theming para que su aspecto se pueda adaptar a los distintos look & feel de las páginas donde los utilicemos.



Miguel Angel Alvarez  
@midesweb

# Crear custom properties

En cualquier lugar de nuestro CSS podemos asignar valores a custom properties.



Miguel Angel Alvarez  
@midesweb

```
body {  
  font-family: sans-serif;  
  background-color: #eee;  
  --my-icon-size: 48px;  
}  
  
.destacar {  
  --my-icon-color: red;  
}  
  
.estrellitas my-icon {  
  --my-icon-color: orange;  
  --my-icon-size: 24px;  
}  
  
.estrellitas my-icon.out {  
  --my-icon-color: #ccc;  
}
```

# Usar custom properties

Dentro del componente podemos asignar custom properties como valores de las propiedades CSS.

```
path {  
  fill: var(--my-icon-color, #888);  
}
```

```
svg {  
  width: var(--my-icon-size, 24px);  
  height: var(--my-icon-size, 24px);  
  display: inline-block;  
}
```



Miguel Angel Alvarez  
@midesweb



# GRACIAS!!



Miguel Angel Alvarez  
@midesweb